DOCUMENT RESUME

ED 165 779 IR 006 837

AUTHOR Heines, Jesse M.
TITLE Evaluating Interactive, Computer-Managed Instruction.
PUB DATE Mar 79
NOTE 13p.; Paper presented at the Annual Conference of the Association for the Development of Computer-based Instructional Systems (San Diego, California, March 1979); For related document, see IR 006 836

EDRS PRICE MF-$0.83 HC-$1.67 Plus Postage.
DESCRIPTORS *Autoinstructional Programs; *Computer Managed Instruction; *Computer Science Education; *Course Evaluation; Flow Charts; Programing Languages; Quality Control; Reliability
IDENTIFIERS *BASIC Programing Language; Computer Software; Digital Equipment Corporation

ABSTRACT

A computer managed instruction (CMI) package to teach BASIC programming, together with questions to be answered in its evaluation, are described. The training package consists of 16 learning modules coupled; an interactive CMI system generates pretests and posttests for each module and branches students to the next appropriate module. It is intended to be sold to customers to be used in a self-paced mode. Evaluation questions pertain to (1) what percentage of customers will use the CMI system, (2) whether students will use the tests as intended, (3) whether a sequential probability test ratio will reduce test length significantly, (4) whether test reliability is maintained, and (5) whether all test items are working properly. (Author/CMV)

EVALUATING INTERACTIVE, COMPUTER-MANAGED INSTRUCTION

Jesse M. Heines

ABSTRACT

This paper describes the evaluation of a new training package on BASIC language programming. The training package consists of 16 learning modules coupled with an interactive Computer-Managed Instruction (CMI) system. The CMI system generates pretests and posttests for each of the modules in the course. The entire training package is intended to be sold to customers to be used in a self-paced mode. This paper discusses the strategies that are being employed to evaluate this approach to the quality control of self-paced instruction.

This paper has been accepted for presentation at the 1979 Annual Conference of the Association for the Development of Computer-based Instructional Systems, San Diego, CA, March 1979.

THE BASIC PRIMER: COMPONENT PARTS

The BASIC Primer is a new self-paced course on BASIC language programming. This course consists of 16 modules which cover topics from simple input/output and computations to the use of sequential and virtual array files. Each module has clearly stated behavorial objectives, text and graphics to instruct the learner on the topics being presented, and exercises to be performed both on paper and on a computer system to reinforce the concepts being presented.

Before each module, students are given the opportunity to take a pretest on that module. If they "pass" this pretest, they are instructed to go on to another module. If they do not pass the pretest, they are asked to study the module and then take a posttest. Students who pass this posttest are routed to the next appropriate module. Students who do not pass the posttest are told the numbers of the objectives on which they missed items, are asked to study these objectives again and do any exercises that they skipped, and then take another posttest. This process is flowcharted in Figure 1.

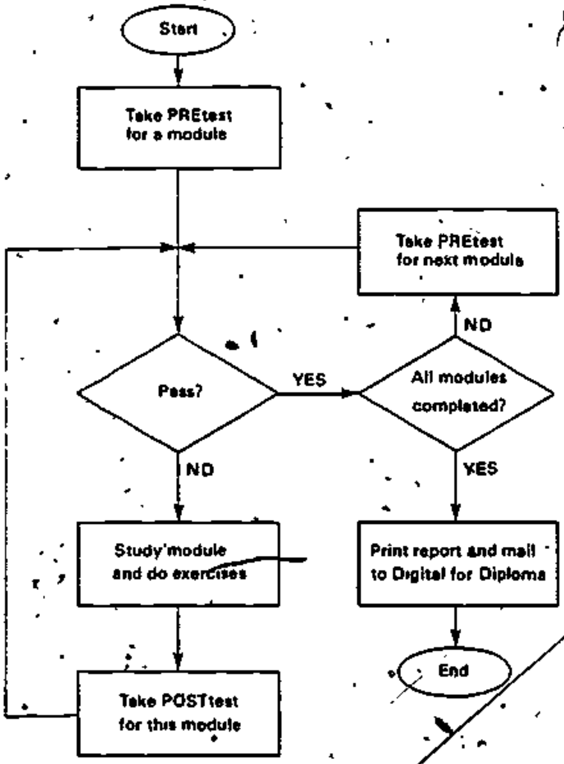

Figure 1

INTERACTION BETWEEN ON-LINE CMI SYSTEM AND OFF-LINE TRAINING PACKAGE

All of the tests for this course are stored on the student's own computer system. This can be a large timesharing system or a small stand-alone system. The software developed for this study is written in a subset of the BASIC language. It can therefore be translated to a variety of operating systems quite easily.

CMI SOFTWARE

The computer-managed instruction system consists of three main programs. These are the Registration Program (CMI), the Router Program (ROUTER), and the Computer-Assisted Test Administration Program (CATSTR). Two other subprograms are also included in this system: the New Student Registration Program (REGSTR) and a Feedback Program (FEEDBK). Students move through these programs as illustrated in Figure 2.

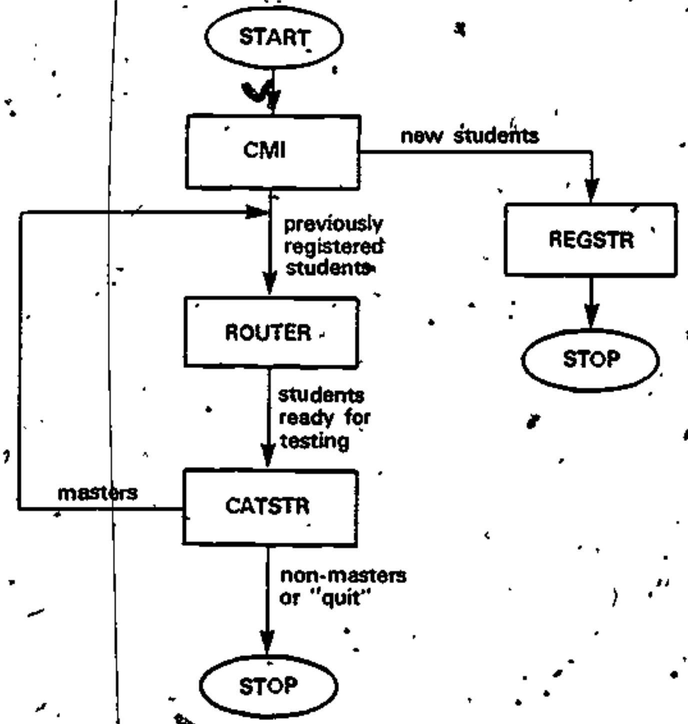

Figure 2

PROGRAM COMPONENTS OF THE CMI SOFTWARE

The Registration Program

The main Registration Program (CMI) does two basic things. First, it asks the user for his or her code name. This uniquely identifies each student so that the data stored on his or her work will be confidential. Second, it uses this code name to search for student-specific data in the roster file. This data includes the student's first and last names, his or her course, and the type of terminal that that student is using. If a student has not yet registered on the CMI system and selected a code name, he or she is directed to press the RETURN key without typing anything else. This causes the system to branch to the New Student Registration Program (REGSTR).

The New Student Registration Program allows students to register themselves on the system. This feature is required because the training package is intended to be used without an instructor. The students enter their terminal types, their names, and their addresses. They are then asked several demographic questions regarding their age and education and motivation for taking the course. This demographic data will allow us to look for trends in student achievement and correlate these with certain student factors.

The New Student Registration Program also produces a text file of the data that it collects from the student. Students are instructed to print this file on paper and mail it back to the course development group. These paper registration forms are kept on file for two reasons. First, it allows the course development group to send out "tickler" letters to students who have not completed the course within a reasonable amount of time. Second, it allows us to get some data on the number of students who start the course but do not complete it.

The Router Program

The purpose of the Router Program (ROUTER) is to identify the test that the student is about to take. This is not always a simple matter. The 16 modules in The BASIC Primer have specific prerequisite relationships. These prerequisite relationships are shown in the module map in Figure 3.

This module map is read in the following manner. First, one begins at the bottom. After completing Module 1, one moves up the map to Module 2. But after completing Module 2, one has the choice of studying either Module 3 or Module 4 because 2 is the only prerequisite to each of these modules. A different kind of relationship exists between Modules 12, 13, and 16. In this case, a student must complete both Modules 12 and 13 before he or she can tackle Module 16.

The Router Program has an algorithm which reads in the prerequisite modules for each of the 16 modules in the course and stores them in an array. This array, combined with the student status on each of the modules in the course, is used to determine for which modules the student has met the prerequisites. If the student has met the prerequisites for only one module, the appropriate pretest or posttest for that module is generated. If a student has met the prerequisites for more than one module, he or she is given the choice of the module to be tested. In this way, the system assures that students go through the course in the manner in which it was intended. This constraint strengthens the instructional design of the course by guaranteeing that each student possesses all of the prerequisite knowledge required for each module that he or she is going to study. The system does not allow a student to take a test on a module for which he or she has not mastered all of the prerequisite modules.

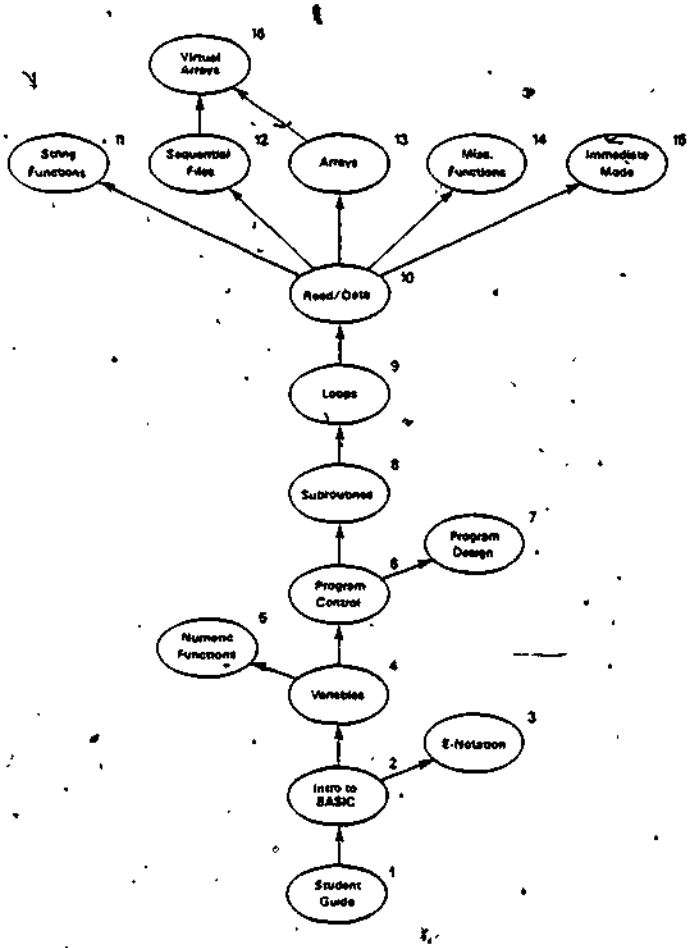

Figure 3

PREREQUISITE RELATIONSHIPS BETWEEN MODULES
IN THE BASIC PRIMER

The Router Program also gives students the option of displaying their statuses on each of the modules in the course. These statuses are reported as:

- Not attempted
- Pretest tried but not completed satisfactorily
- Pretest completed satisfactorily; posttest skipped
- Posttest tried but not completed satisfactorily
- Posttest completed satisfactorily

Students are allowed to take a pretest for a module only once. All subsequent tests are automatically interpreted as posttests, because it is assumed that if a student does not pass a pretest he or she will go and study the corresponding module. Students are allowed to take posttests as many times as necessary to complete them satisfactorily, i.e., demonstrate mastery on all of the objectives in that module. (Tests are generated interactively in real time, so no two tests are exactly alike. See the discussion below.)

The Testing Program

The Computer-Assisted Test Administration Program (CATSTR) generates both pretests and posttests. These tests are administered to students at a computer terminal. The purpose of the Computer-Assisted Test Administration Program is to classify the student as either a master or a non-master on the specific module being tested, and to make this classification in a minimum amount of time. To accomplish this, the Test Administration Program evaluates each student response with a sequential probability test ratio. This algorithm is discussed briefly below, but has been described elsewhere in detail (Heines, 1978a and 1978b).

The algorithm evaluates a student's score after each item is presented. This evaluation is used to classify the student in one of three categories. If the student's score exceeds a mastery threshold with a predefined level of confidence, the student is classified as a master and testing is terminated. If the student's score is less than a non-mastery threshold with a specified degree of confidence, the student is classified as a non-master. If the student's score falls within these two thresholds, another item is presented. Once the student responds to this item, the evaluation algorithm is repeated.

Using the sequential probability test ratio, it is possible for a student to remain forever in the "uncertainty band" between the mastery and non-mastery thresholds. For this reason, testing is terminated after 30 items have been presented regardless of the student's score. At this point, the student is classified as either a master or a non-master by determining the threshold that his or her current score is closest to.

Thirty is a rather large number of items for each module. To make the tests truly repeatable, more than 30 items are needed in the item bank for each module so that the randomized selection process does not produce identical tests. For The BASIC Primer, there are 754 items in the entire bank. This allows an average of about 45 items for each module. Since it is expected that very few tests will actually be 30-items long, this average is sufficient to yield a very large number of different test forms.

Students who demonstrate mastery on a test are branched back to the Router Program. They then select the next module on which they would like to be tested. Students who do not demonstrate mastery on a test are told the objectives on which they missed items. The CMI system then halts, and students are directed to do additional study off-line.

Students who have completed all of the modules in the course are branched to the Feedback Program. This program asks for student comments on the course and generates another data listing file. This listing file verifies that the student has earned a diploma and contains other data that is needed by the course development group to evaluate the training package. Students are directed to mail this file back to the course development group either on a machine-readable media, or they are asked to print this file on paper and mail it. In either case, this data is combined with data from other students in other training locations by a large system within the course development group. It is this feedback data that is used to evaluate the training package as a whole.

EVALUATION QUESTIONS AND STRATEGIES

The data described in the preceding section will be used to try to answer five research questions on the system. Each of these questions and the strategy that will be used to evaluate the data is described in the paragraphs that follow.

(1) What percentage of customers who study the training materials will use the CMI system?

There are two ways in which we will know how many students are using The BASIC Primer. First, we will have data on the number of CMI systems in use from our corporate sales data. Second, we will have some idea of the number of students taking the course by the registration forms that are generated by the New Registration Program. These two pieces of data, taken together, should give us a good idea of the number of people who at least began using the system.

After any student has completed the first ten modules of the course, a new listing file will be generated for that student. This file will contain the status data on all of the students who are registered on that system. From this data we will be able to ascertain how many students began the course but did not complete it. That is, we should be able to know how many students began using the testing system and did not continue. If we find that a large number of students fall into this category, it will be necessary for us to investigate further the reasons that they discontinued the use of the CMI system. Most importantly, we would like to know whether they continued studying the modules but just gave up studying the tests. If this is the case, it will be clear that the CMI system did not fulfill its goal. On the other hand, if we find that most students did continue to use the CMI system, the status data generated at this point will give us an excellent picture of how the system and mastery algorithm are working.

(2) Will those students who use the CMI system use the tests as intended?

In addition to updating the status records for a particular student each time that he or she takes the test, the system will maintain a test history file. This file will record data on each test that is administered in chronological order. Each test history record will record

- the student number,
- the module being tested,
- whether the test being administered is a pretest or a posttest,
- whether the test being administered is a normal test or a special 30 item test (used for reliability measurements; see below), and
- the date and time that the test was started.

This data will allow us to ascertain whether a student is just retaking tests one right after the other or actually studying between test administrations. We will assume that if a student repeats a test in less than ten minutes, that he or she did not return to the materials before requesting a retest. On the other hand, if we find that the time between repeated tests is on the order of 15 to 30 minutes, we will assume that the student did look over the material and possibly do additional exercises before he or she returned to the system for a retest.

The test history file will also tell us the order in which students go through the course modules. It will provide a trail which we can analyze to assure that the Router Program did not allow students to go on to more advanced modules before they met the prerequisites.

(3) Does the sequential probability test ratio reduce test lengths significantly?

Each module on the system has an associated log file. One of the types of data that is stored in this log file is the length of each of the tests that is taken. These lengths are recorded in four categories. They are recorded for masters on pretests, non-masters on pretests, masters on posttests, and non-masters on posttests. If the sequential probability test ratio works properly, we should find that the lengths of non-master tests are significantly less than the lengths of master tests. In addition, we should find that pretests on which mastery decisions were made are, on the average, longer than posttests on which mastery decisions are made. This result should occur because the criteria for mastery on pretests are more stringent than the mastery criteria on posttests. In addition, we hope to find that the average length of a test is considerably shorter than 30 items. This will indicate that the sequential probability test ratio was useful in reducing test lengths.

Two additional items that are stored for each student will help us ascertain whether any reduction in test length due to the sequential probability test ratio is significant. These items are the total amount of time that each student spent testing, and the total number of test items that were presented to that student. By dividing the total time by the total number of items, we will have an indicator of the average amount of time that it takes for a student to respond to a single test item. By looking at the number of tests that the student has taken and the amount of time that would have been required if these had been fixed length tests of, say, 20 or 30 items, we will have an indicator of the amount of time saved by using the sequential probability test ratio. Again, the decision algorithm will prove itself useful if the amount of time saved is a significant portion of the total amount of time spent testing.

(4) Is test reliability maintained even when the sequential probability test ratio causes short tests to be administered?

To obtain a measure of the tests' reliability, every fifth test that is administered to a student will be forced to 30 items long regardless of the results of the sequential probability test ratio. When this occurs, the system will record two decisions for the student. It will first record the decision that it would have made if testing had been terminated as soon as possible by applying the sequential probability test ratio. It will then continue the test and record the decision that is made after 30 items have been presented. These two decisions will be compared in a 2 x 2 contingency table. The analysis of this contingency table will be made by applying the G index. This index is a refinement by Livingston (1976) of the percentage of agreement first described by Carver (1970).

		CLASSIFICATION ON T1	
		Master	Non-Master
CLASSIFICATION ON T2	Master	a	c
	Non-Master	b	d

Figure 4

FREQUENCIES OF AGREEMENT BETWEEN MASTERY AND NON-MASTERY CLASSIFICATIONS ON TWO SETS OF TEST DATA

Using the contingency table in Figure 4, Carver defined the percentage of agreement as:

$$P0 = \frac{a+d}{a+b+c+d}$$

This measurement varies between 0 and 1.

Swaminathan et al. (1974) agreed with Carver's concept of criterion-referenced reliability as a coefficient of classification, but preferred a more sophisticated computation known as the kappa coefficient. Swezey and Pearlstein (1975) preferred an even more sophisticated approach known as the phi coefficient. Livingston's coefficient, the G index, is simply:

$$G = 2 \times (P0 - 0.5)$$

This coefficient varies between -1 and +1, and is therefore compatible with the kappa and phi coefficients. (A more detailed discussion of this mathematics is provided in Heines, 1974b.)

(5) Are all of the test items on the system working properly?

In addition to the test length data, the log file for each module also records item analysis data. This data is a tally of the number of times of that each response was chosen for each question presented. Like the test length data, the item analysis data is separated into four catagories. That is, tallies are kept separately for

- pretests on which mastery decisions are made,
- pretests on which non-mastery decisions were made,
- posttests on which mastery decisions were made, and
- posttests on which non-mastery decisions were made.

This allows us to compute pretest/posttest and master/non-master discrimination indices. Therefore, we will be able to ascertain how well each item discriminates between masters and non-masters as well as how well it discriminates between pretests and posttests. The log data from various training sites will be combined to give us a large data sample. This should make the item analysis very significant.

GLOBAL ISSUES

The BASIC Primer is the first computer-based course being offered by Digital Equipment Corporation. We feel that the use of the computer in our current mode gives us the power to evaluate our training materials objectively. Never before have we been able to collect this type of data on a training course. If the CMI system fulfills its goals, it will not only improve the quality control of our courses, but it will provide us with a vehicle for improving all of the courses that we offer in this mode. The BASIC Primer and its complementary CMI system are currently being tested in selected sites in New England. The final version of this training package should be available to customers in the spring of 1979.

REFERENCES

CARVER, R. P. Special problems in measuring change with psychometric devices. In Evaluative Research: Statistics and Methods. American Institute for Research, Washington, D.C. 1970.

HEINES, Jesse M. The use of computer-managed instruction to control on-site, self-instructional training in a small systems customer environment. Proceedings of the 1978 conference of the Association for the Development of Computer-Based Instructional Systems, Dallas, Texas. March, 1978a.

__________. The use of interactive, computer-managed instruction to control the quality of self-paced training without requiring the presence of an instructor. Proceedings of the 1978 conference of the Association for Computing Machinery, Washington, D.C., December 1978b.

LIVINGSTON, Steve. Lecture on "statistical concepts in criterion-referenced measurement". Educational Testing Service, Princeton, N.J. February 23, 1976.

SWAMINATHAN, H., R.K. Hambleton, and J.J. Algina. Reliability of criterion-referenced tests: a decision-theoretic formulation. Journal of Educational Measurement 11:263267. 1974.

SWEZEY, R.W., and R.B. Pearlstein. Guidebook for Developing Criterion-Referenced Tests. US Army Research Institute for the Behavioral and Social Sciences. 1975.